Wu's methodの浮動小数化 (数式処理における理論と応用の研究)

AUTHOR(S):

野竹, 禎雄; 甲斐, 博; 支, 麗紅; 野田, 松太郎

数理解析研究所講究録
1199巻 2001年 1-9

Wu's method の浮動小数化

愛媛大学工学部 野竹 禎雄(Yoshio NOTAKE) *
愛媛大学工学部 甲斐 博(Hiroshi KAI) †
愛媛大学工学部 支 麗紅(Lihong ZHI) ‡
愛媛大学工学部 野田 松太郎(Matu-Tarow NODA) §

1 はじめに

連立代数方程式の解を高精度で求めることは、応用数学分野の長年の夢の一つであった。1変数代数方程式にはニュートン法に基づく高精度の数値計算法が広く用いられているが、多変数連立代数方程式に対しては、連続変形法が知られている程度である。一方で、多変数連立代数方程式を解く必要のある問題は、ロボット工学、計算機支援設計、制御工学などの分野で多く提出されつつある。これらの問題の特色は、

1. 問題が大規模のものが多く、方程式が良条件か否かの判定や解の挙動がわからない。

2. 構成する代数方程式の係数が浮動小数であったり、パラメータであったりする。解が近接しているか否かや、解が0次元であるか否かさえ不明な場合が多い。このため、この種の問題に、無批判的に数値計算で解を求めようとするのはきわめて危険である。

そこで、多変数連立代数方程式の解を記号計算で求めようという試みがなされている。与えられた多変数連立代数方程式は記号的に三角化され、高精度に求めた1変数代数方程式の解を順に代入し、最終的にすべての解を求める。方程式の三角化に用いられる方法は、グレブナ基底によるものが中心であり、その高速化も研究されている [1]。また、ascending set を取り出すことにより三角化を達成する characteristic set の方法 [2](以下、Wu's method と呼ぶ) が有力であり、これらの優劣はまだ定まっていない [3]。

---

*notake@hpc.cs.ehime-u.ac.jp
†kai@cs.ehime-u.ac.jp
‡lzhi@hpc.cs.ehime-u.ac.jp
§noda@cs.ehime-u.ac.jp

Wu's method の数式処理システムへのインプリメントは Maple 上で行われている [4]。本研究では、これをを改良し、Risa/Asir にインプリメントする。さらに、連立代数方程式の解が必要な問題への対応を考え、浮動小数係数の場合にも、Wu's method による三角化を可能なように拡張する。この場合、Wu's method の計算途中に必要な擬除算を行う場合に、近似 GCD[5] の導入が必要になる。以下、まず Wu's method の Risa/Asir へのインプリメント及びグレブナ基底によるものとの比較を行う。次いで、Wu's method の浮動小数化を２つの方法で行う。その第一は、単純に擬除算部分に近似 GCD を導入したものであり、第二は安定化理論 [6] による浮動小数化である。最後に、両者の比較などを行う。

2 Wu's method のアルゴリズム

Wu's method は２つのブロック (basset,remset) に分けられる。これらの過程を交互に反復し、以下のように連立代数方程式を三角化する。

$$\left\{\begin{array}{l} f_1(x_1,x_2,\ldots,x_n)=0 \\ f_2(x_1,x_2,\ldots,x_n)=0 \\ \quad\vdots \\ f_n(x_1,x_2,\ldots,x_n)=0 \end{array}\right. \quad\rightarrow\quad \left\{\begin{array}{l} \tilde{f}_1(x_1)=0 \\ \tilde{f}_2(x_1,x_2)=0 \\ \quad\vdots \\ \tilde{f}_n(x_1,x_2,\ldots,x_n)=0 \end{array}\right.$$

入力 (多項式集合)　　　　出力 (特性多項式集合)

ここで、PS, CS, RS は多項式集合である。その他の記号は、

- lvar(p_i),ldeg(p_i),ini(p_i):各々 p_i の主変数、主変数の次数、主係数
- ascending set:以下の条件を満たす多項式集合
  - lvar(p_1) $<$ lvar(p_2) $<\cdots<$ lvar(p_n)
  - p_i における lvar(p_i) の次数$<$ p_j における lvar(p_i) の次数の時 $i<j$

である。Wu's method のアルゴリズムは次ぎのように書ける。

アルゴリズム 1 (Wu's method)
入力: 多項式集合 PS
出力: Characteristic Set CS

step1 $CS \leftarrow$ *basset(*PS*)* : PS から *ascending set* を取り出す

step2 $RS \leftarrow$ *remset(*PS,CS*)*

1. RS を空集合とする.

2. 各 $r_i = p_i \in PS$ に対し次の式を求める.
   - 各 $c_j \in CS$ に対し $r_i = prem(r_i, c_j)$ を求める.
   - 得られた r_i を RS に加える. $RS = RS \cup \{r_i\}$

ここで, *prem* は擬除算による擬剰余を表す.

step3 零判定： *if*(RS が空集合) *then* *Return* CS
else $PS = PS \cup RS$
step 1 に戻る

3 Wu's method のインプリメントとグレブナ基底計算

Wu's method の Risa/Asir へのインプリメントは、Maple 上でのインプリメント [4] を参考にして行った。数式処理システムの相違による問題点に対しては、以下のように対応した。

1. 多項式の表現方法

Maple では単項式ごとの和で表現するが、Risa/Asir では主変数の次数により表現する。単項式ごとの表現が可能な分散表現多項式に変換したうえで処理を行う

2. いくつかのシステム固有の関数

集合 $expr$ の i 番目の要素を取り出す関数 op($i, expr$) 等の関数は、新しく作成し直した。

インプリメントした Wu's method を Risa/Asir 上でグレブナ基底によるものと、計算速度の面で比較した。例１１、２の連立代数方程式は、いずれも０次元の解を持たず、有限次元の解を持つものである。なお、計算は DEC ALPHA 433MHz Memory:128MB によった。

例１： order$[a, b, c, d, e, f]$

$$\begin{cases} ab + ce + f + 4 = 0 \\ a + bc + c^2 + d + e + f^2 = 0 \\ a + a^2 + b^3cf + cd + ce = 0 \\ 3a + 3b + b^2f + 4cd^2 + e^2 + 12 = 0 \end{cases}$$

例２： order$[a, b, c, d, e, f, g]$

$$\begin{cases} a + b + c + d + e + f + g = 0 \\ a + cd + e^2 + 4bg + 12 = 0 \\ fg + a^3 + a^2b + cd + ed = 0 \\ 3a + b + 3dg + 4c + e * c + 12 = 0 \end{cases}$$

表１：計算時間の比較 (単位：sec)

	Wu's method	グレブナ基底計算
Example1	4.32(+1.81)	>200
Example2	19.13(+12.22)	>200

以上より、本論の Wu's method のインプリメントは正しく行われており、特に有限次元の解を持つ連立代数方程式のいくつかにつては、グレブナ基底によるものより、はるかに高速に三角化を行うことが可能であることがわかった。

4 浮動小数での Wu's method

浮動小数係数の連立代数方程式を Wu's method を用いて三角化することを考える。整数や有理数のような正確な係数の場合に対して上で述べた Risa/Asir 上へのインプリメントを浮動小数のような不正確な係数を持つ連立代数方程式の場合へ拡張するためには、Wu's method のアルゴリズムにおける擬除算中の GCD 計算を近似 GCD 算法 [5] に置き換え、微小項をしきい操作で無視する方法と、各係数をその中心値と許容誤差による円盤型区間数であるブラケット係数にし、この係数の書き換え規則を利用し、計算桁数を増加させつつ正しい結果への収束を試みる安定化理論 [6] による方法とである。ここでは、これらを Risa/Asir にインプリメントし比較する。なお、インプリメントに際し、係数の区間数は矩形型区間数を用いた。したがって、安定化理論におけるブラケットに対応する区関数は、区間の上限と下限で示している。正確な代数計算とは異なり、このような数値数式融合のハイブリッド計算では、浮動小数係数のようなある種の近似計算における計算誤差の評価を行う必要がある。そこで、まず Wu' method の浮動小数化による誤差評価を行う。

4.1 浮動小数 Wu's method の誤差評価

浮動小数係数の場合に、誤差を考慮すべき点は step2 の擬除算においてである。Wu's method で用いる擬除算は

$$I^{1+\mathrm{ldeg}(f_1)-\mathrm{ldeg}(f_2)} f_1 = f_2 Q + R$$

と書ける。ここでの各記号は以下を表す。

- $f_1, f_2 : X = (x_1, \cdots, x_n)$ の正確な係数の浮動小数係数多項式
- I,Q,R : 各々 $\mathrm{ini}(f_2)$、擬商、擬剰余

このとき、不正確係数の多項式 $F_1(X), F_2(X)$ を考えると

$$F_i(X) = f_i(X) + \epsilon_{f_i}(X), \qquad i = 1, 2$$

であり、$\epsilon_{f_i}(X)$ は F_i に含まれる微小誤差項である。

このとき、F_1, F_2 の擬除算は以下のように示すことができる。

$$\tilde{I}^{1+\mathrm{ldeg}(f_1)-\mathrm{ldeg}(f_2)} F_1 = F_2 \tilde{Q} + \tilde{R}$$

ここで、$\tilde{I}$ は $\tilde{I} = \mathrm{ini}(F_2) = I + \mathrm{ini}(\epsilon_{f_2})$ と表される。

擬剰余の誤差評価 δR を $\delta R = ||\tilde{R} - R||$ とする。２つの入力多項式の差、$1 + \mathrm{ldeg}(f_1) - \mathrm{ldeg}(f_2)$ が十分小さく、以下を満たす場合を考える。

$$||f_1|| \approx ||f_2|| \approx ||Q|| \approx 1, \qquad ||\epsilon_{f_1}|| \approx ||\epsilon_{f_2}|| \approx ||\tilde{Q} - Q|| \approx \epsilon$$

この場合、誤差 $||\delta R||$ は、$||\delta R|| \approx \epsilon$ となる。以下に例によって上の評価の正しさ示す。

例 3

$$\left\{\begin{array}{lll} F_1 & = & (1+\epsilon_{11})x^2 + (1+\epsilon_{12})z + (1+\epsilon_{13}) \\ F_2 & = & (1+\epsilon_{21})xy + (1+\epsilon_{22}) \\ F_3 & = & (1+\epsilon_{31})x + (1+\epsilon_{32})z \end{array}\right.$$

$\mathrm{prem}(F_1, F_3)$ に着目する。このとき $I = \mathrm{ini}(F_3) = 1, \quad 1 + \mathrm{ldeg}(F_1) - \mathrm{ldeg}(F_3) = 1$ となり上の条件を満たす。$||\epsilon_{ij}|| = O(\epsilon)$ とすると，$F_4 = x^2 - x + 1 + O(\epsilon)$ となり、誤差 $||F_4 - f_4|| = O(\epsilon)$ となる。

ここで、$\mathrm{ini}(f_1)$ と $\mathrm{ini}(f_2)$ が最大公約数 (GCD) を持つ場合の擬除算を考える。最大公約数を持つ場合、擬除算内で近似 GCD を用いるので、そこでの誤差評価も行う必要がある。ここではまず係数が正確な場合について考える。このとき以下の関係式を導くことができる。

$$g = \mathrm{GCD}(\mathrm{ini}(f_1), \mathrm{ini}(f_2)), \qquad \mathrm{ini}(f_1) = \tilde{f}_1 g, \qquad \mathrm{ini}(f_2) = \tilde{f}_2 g.$$

ここで、$\tilde{f}$ は GCD との積が $\mathrm{ini}(f)$ となる多項式を表す。以上から主係数 I は

$$I = \mathrm{ini}(f_2)/g = \tilde{f}_2$$

となる。次ぎに係数が不正確な場合について考える。このとき、GCD は精度が ϵ の Ochi et.al. による多変数近似 GCD[5] を用いる。上と同様に次式を得る。ここでの、$\mathrm{apxGCD}(\mathrm{ini}(F_1), \mathrm{ini}(F_2); \epsilon)$ は浮動小数係数の多項式 P,Q の精度 ϵ の近似 GCD を表す。

$$\left\{\begin{array}{rcl} G & = & \mathrm{apxGCD}(\mathrm{ini}(F_1), \mathrm{ini}(F_2); \epsilon) \\ \mathrm{ini}(F_1) & = & \tilde{F}_1 G + \delta\,\mathrm{ini}(F_1), \qquad \mathrm{ini}(F_2) = \tilde{F}_2 G + \delta\,\mathrm{ini}(F_2), \\ ||\delta\,\mathrm{ini}(F_1)|| & = & O(\epsilon), \qquad ||\delta\,\mathrm{ini}(F_2)|| = O(\epsilon) \end{array}\right.$$

以上から浮動小数係数等不正確な係数の場合は、主係数 $\tilde{I}$ は次のようになる。

$$\tilde{I} = \mathrm{ini}(F_2)/G = \tilde{F}_2$$

ここで、$||g|| \approx 1$ かつ $||G - g|| \approx \epsilon$ という条件の場合を考えると、最大公約数を持つ場合の主係数の誤差は以下のようになる。

$$||I - \tilde{I}|| = ||\tilde{f}_2 - \tilde{F}_2|| \approx \epsilon$$

4.2 Wu's method の近似 GCD の導入

入力の多項式集合の係数を浮動小数にした場合に、２で述べた Wu's method のアルゴリズムのインプリメントにおいて注意すべき点は、

1. 関数等を浮動小数を許すように書き直す。

2. 剰余多項式の集合 RS を求める場合に、remset(PS,CS) の計算に注意する。

1. の修正は当然であるが、2. に関しては、擬除算の演算で浮動小数を許すと、正確な GCD を近似 GCD で置き直す必要が生じる。そこで、アルゴリズムは次ぎのようになる。

アルゴリズム 2 (浮動小数係数の Wu's method)
入力: 多項式集合 PS, ϵ
出力: Characteristic Set CS

step1 $CS \leftarrow$ *basset(*PS*)* : PS から *ascending set* を取り出す

step2 $RS \leftarrow$ *apx-remset(*PS,CS;ϵ*)*

1. RS を空集合とする
2. 各 $r_i = p_i \in PS$ に対し次の式を求める
   - 各 $c_j \in CS$ に対し $r_i =$ *apx-prem*$(r_i, c_j; \epsilon)$ を求める
   - 得られた r_i を RS に加える $RS = RS \cup \{r_i\}$

ここで，*apx-prem* は近似 *GCD* を用いた擬除算を表す

step3 零判定 : *if(*RS の要素 r_1 が $||r_i|| < \epsilon$ を満たす*)* *then* *Return* CS
else $PS = PS \cup RS$
step 1 に戻る

アルゴリズム 1 と比較すると、step2 での GCD 演算が近似 GCD になったいる他、step3 での零判定がしきい値 ϵ のしきい操作になっていることに注意する。

実際に浮動小数係数の Wu's method を用いた計算例を以下に示す。

例 4: $\epsilon = 0.001$, 変数順序 $x \prec y$

$$\begin{cases} F_1 &= (x-y)(x+1)(x^2+y^2-1.001) \\ &= (x-y)(x+1)(x^2+y^2-1) + 0.001g(x,y) = f_1 + 0.001g(x,y) \\ F_2 &= (x-y)(x^2+y^2-4) = f_2 \end{cases}$$

出力 : Characteristic Set CS

$$\begin{aligned} \text{不正確な入力 : charsets}(F_1, F_2) &= -2.999x^2 + (2.999y - 2.999)x + 2.999y \\ \text{正確な入力 : charsets}(f_1, f_2) &= -3x^2 + (3y-3)x + 3y \end{aligned}$$

このときの誤差は次のようになる。

$$\text{charsets}(F_1, F_2) - \text{charsets}(f_1, f_2) = -0.001(x^2 - (y-1)x - y) = 0.001g(x,y).$$

4.3 安定化理論による Wu's method の浮動小数化

上で述べたような、GCD 演算を近似 GCD 演算で置き換え、零判定操作をしきい操作で対処する方法は、浮動小数係数の連立代数方程式に対する三角化を可能にする。しかし、しきい値選択のあいまいさが残されているので、次ぎに安定して零判定が行えるよう、安定化理論 [6] を用いて Wu ' s method の浮動小数化を行う。安定化理論を用いると Zero-Rewriting により、多項式の次数が大きくなり誤差が動的に変化しても零判定を安定して行うことが可能となる。

なお、インプリメントにおいては、多項式の係数を矩形型区間数で表現している。剰余多項式 RS を求める step2 において、apx-remset, apx-prem 等を区間数演算による interval-remset, interval-prem 等に置き換えている。interval-prem においては、区間数係数の近似 GCD を求める必要がある。この概念と計算の実際は、文献 [7] にあり、多項式剰余列が 0 になり計算が停止するべき場合には、0 を含む区間数が求まることが示されている。安定化理論は、この点をさらに理論的に正当化している。

安定化理論の Wu's method のアルゴリズムは以下の通りである。

アルゴリズム 3 (安定化理論の Wu's method)
入力: 多項式集合 PS(区間数係数)
出力: Characteristic Set CS

step1 $CS \leftarrow$ *basset*(PS) : PS から *ascending set* を取り出す

step2 $RS \leftarrow$ *interval-remset*(PS,CS)

1. RS を空集合とする
2. 各 $r_i = p_i \in PS$ に対し次の式を求める
   - 各 $c_j \in CS$ に対し $r_i =$ *interval- prem*(r_i, c_j) を求める
   - 得られた r_i を RS に加える $RS = RS \cup \{r_i\}$

ここで, *interval-prem* は区間数の擬除算を表す

step3 零判定 : *if*(RS が空集合) *then* *Return* CS
else $PS = PS \cup RS$
step 1 に戻る

なお、step2 内の擬除算部での GCD を近似 GCD をさらに区間数の入力に変更したものに置き換え、step3 での零判定は Zero-Rewriting により誤差の処理を行っている。

実際に安定化理論の Wu's method を用いた計算例を以下に示す。

例 5

入力多項式 $(x \prec y \prec z)$

$$PS = \begin{cases} p_1 = [0.999, 1.001]x^2 + [0.998, 1]z + [1, 1.001] \\ p_2 = [0.999, 1.001]xy + [0.999, 1] \\ p_3 = [1, 1.001]x + [0.999, 1]z \end{cases}$$

step1 Basset $CS = \{p_2, p_3\}$

step2 Remset

prem(p_1, p_3)
$= [0.998001, 1.001]x^2 - [0.998, 1.001]x + [0.999, 1.001] = p_4$
prem$(p_4, p_2) = p_4$
$RS \leftarrow p_4$

step3 零判定 Remset(PS, CS) は零でない $\rightarrow PS = \{p_1, p_2, p_3, p_4\}$, step1 へ

出力 Characteristic Set CS

$\{[0.998001, 1.001]x^2 - [0.998, 1.001]x + [0.999, 1.001],$
$[0.999, 1.001]xy + [0.999, 1], [1, 1.001]x + [0.999, 1]z\}$

5 むすび

以上述べてきたように、本論では Wu ’s method を Risa/Asir にインプリメントし、連立代数方程式を安定に解くために、多項式集合の三角化を行った。最終的に三角化された多項式中に出現する 1 変数多項式の解は、良条件の場合は問題なく高精度で数値的に求めることが可能であり、悪条件の場合にも、近似 GCD 演算等を経由して求めることができる。特に、解が 0 次元でない有限次元の場合は、グレブナ基底ではうまく計算できない場合にも、十分な速度で三角化が可能であることを具体的な例を通して示した。次に、二通りの方法で、浮動小数係数の連立代数方程式に対する Wu ’s method をインプリメントし、比較した。それらは、Wu ’s method で剰余多項式を求める場合に

1. 擬除算演算を近似 GCD で置き換え、かつ零判定をしきい操作によるもの
2. 擬除算演算を区間数係数の近似 GCD で置き換え、零判定は桁数を増加させながら反復計算する安定化を採用するもの

である。連立代数方程式が大規模になると、しきい操作の場合は、厳密にしきい値を設定することが困難になる一方、安定化理論を用いると零判定は問題なく行える。浮動小数係数の連立代数方程式を高精度で安定して解く必要は、今後ますます増加するものと思われる。本論の方法を、さらに高速化することや、誤差評価の一層の進展等が必要な課題である。

参 考 文 献

[1] J.C.Faugere,http://calfor.lip6.fr/ jcf/Software/index.html

[2] W.-t. Wu:A Mechanization method of equation-solving and theorem-proving,*Advances in Computing Research*,**vol.6**,pp103-138,1992.

[3] D.Lazard:Resolution of Polynomial System,*Proc.ASCM2000*,pp1-8,2000

[4] D. Wang:Implementation and Applications of Charcteristic Set Method ,*Lecture Notes for 1994 Summer Graduate School in Mathmatics*,**Preliminary Version**,1994.

[5] M.Ochi,M.-T.Noda and T.Sasaki :Approximate GCD of multivariate polynomials and application to ill-conditioned system of algebraic equations, *J. Inf. Proc.*,**vol.14**,pp.292-300,1991.

[6] K.Shirayanagi and M.Sweedler:A Theory Stabilizing Algebraic Algorithms, *Tech.Rep.95-28*,**Cornell Univ**,pp1-92,1995.

[7] M.-T. Noda and T.Sasaki:The interval arithmetic for the ill-conditioned polynomial equation , 数理解析研講究録,**No.673 「自己検証的計算法とその応用」**,pp47-61,1988.